Integrating Machine Learning Techniques in a Guided Discovery Tutoring Environment: MEMOCAR

JEAN-DANIEL ZUCKER
LAFORIA-IBP, Université Paris VI
4, Place Jussieu - Boîte 169, 75252
PARIS Cedex 05, FRANCE

Abstract This chapter presents how Machine Learning Techniques can effectively contribute to improve the quality of interactions in Guided Discovery Tutoring Environments (GDTE) . We review several approaches to integrate Machine Learning in ITS. Most of these approaches use concept learning from examples to maintain a Student Model. We go along presenting an alternative use of induction techniques to learn concepts on the same data that are presented to the learner. We present on a concrete example how this approach is integrated in a GDTE called MEMOCAR, a Computer Aided Language Learning System for Chinese characters. Three main types of activity are identified in MEMOCAR: familiarization with Chinese characters, collaborative discovery of similarities between characters and exercises to test characters acquisition. The stage of familiarization is supported by exploration of hyperdata whilst collaborative discovery and exercises' diagnosis are supported by a tool based on CHARADE, a top-down induction system. Such integration offers a new alternative to the complex problem of making Guided Discovery Tutoring Environment more collaborative.

Résumé: Le thème abordé dans ce rapport est celui de l'utilisation de techniques d'apprentissage symbolique automatique (ASA) pour l'amélioration des interactions dans les environnements d'enseignement par la découverte assisté par ordinateur. Nous présentons différents environnements intégrant des techniques d'apprentissage symbolique automatique; principalement pour construire et/ou mettre à jour un modèle de l'élève. Nous présentons ensuite une intégration originale de l'ASA dans le cadre d'un environnement dédié à l'apprentissage des caractères chinois par la découverte: MEMOCAR. Notre approche consiste à utiliser l'ASA pour faire apprendre au système sur les mêmes données que l'élève, et à utiliser les résultats de cet apprentissage pour construire des interactions. Deux des principales activités du système MEMOCAR, la découverte par l'apprenant de similarités entre caractères et les exercices de test de l'apprentissage sont ainsi basées sur l'adaptation d'un algorithme inductif: CHARADE. Ce type d'intégration de l'ASA dans les environnements d'apprentissage par la découverte s'insère dans les recherches visant à rendre de tels environnements plus coopératifs.

To be published as a chapter in a book entitled "Emerging Technologies"

Integrating Machine Learning Techniques in a Guided Discovery Tutoring Environment: MEMOCAR

JEAN-DANIEL ZUCKER
LAFORIA-IBP, Université Paris VI
4, Place Jussieu - Boîte 169, 75252
PARIS Cedex 05, FRANCE

Abstract This chapter presents how Machine Learning Techniques can effectively contribute to improve the quality of interactions in Guided Discovery Tutoring Environments (GDTE) . We review several approaches to integrate Machine Learning in ITS. Most of these approaches use concept learning from examples to maintain a Student Model. We go along presenting an alternative use of induction techniques to learn concepts on the same data that are presented to the learner. We present on a concrete example how this approach is integrated in a GDTE called MEMOCAR, a Computer Aided Language Learning System for Chinese characters. Three main types of activity are identified in MEMOCAR: familiarization with Chinese characters, collaborative discovery of similarities between characters and exercises to test characters acquisition. The stage of familiarization is supported by exploration of hyperdata whilst collaborative discovery and exercises' diagnosis are supported by a tool based on CHARADE, a top-down induction system. Such integration offers a new alternative to the complex problem of making Guided Discovery Tutoring Environment more collaborative.

The limitations of environments solely based on discovery through unconstrained exploration are well known (Elsom-Cook, 1990). This is especially true for domains where activities cannot be solved by a unique “good solution” that can be translated into a simple rule and easily explained and taught. On the other hand, in such domains, and within a framework of a constructionist theory of knowledge representation (Papert, 1991), an approach where the learner is too guided appears inefficient because it is too constraining.

In this way, adequate environments need to support a collaboration between the learner and the system (Gilmore & Self, 1988). We support the idea that Machine Learning Techniques (MLT) may improve collaboration between the learner and the system by providing means to efficiently individualize *interactions*. In such interactions, the system may be seen by the learner as contributing to solve a "problem" that has been selected either by the system or by the learner, depending on the teaching style.

In this chapter we shall begin with a brief overview of Machine Learning Techniques integrated in Intelligent Tutoring Systems (ITS). We shall then concentrate on the description of the use of inductive techniques to provide the learners with more *individualized* interactions in GDTE. We finally present a real integration of such technique in a Computer Aided Language Learning Systems for Chinese characters called MEMOCAR.

COMPARING APPLICATIONS OF MACHINE LEARNING IN ITS

Within ITS, the individualization of the interaction learner/system is made possible by a specific component: the Student Model (see R. Mizoguchi's section elsewhere in this book). This component is defined so as to reflect the current state of the learner knowledge. A Student Model is principally used to individualize instruction: choose the next topic to be taught, generate new problems, adapt explanations, etc. The main reason for having integrated MLT to ITS is to build and maintain Student Models. Among the numerous techniques developed in the framework of Machine Learning, *concept learning from examples* is the one that has been mostly used when integrated in ITS. So as to compare these systems, we shall represent the induction problem by the search of a set of hypotheses (*Hyp*) that, along with background knowledge (*BK*) allows the system to deduce some observed concepts or facts (*Obs*). The induction process may then be expressed as to find *Hyp* that satisfies the entailment $Hyp \cup BK \models Obs$ (Michalski, 1983; Michalski, 1991)

Machine Learning to maintain Student Models

The first approach used for integrating MLT within ITS is related to *maintaining the Student Model* (Gilmore & Self, 1988; Woolf & Murray, 1994). In PIXIE (Sleeman, 1983), ACM (Langley & Ohlson, 1984), THEMIS (Kono, 1993) or ELECTRE (Paliès, Caillot, Cauzimille-Marmèche, Laurière, & Mathieu, 1986), the Student Model is viewed as “procedures for problem solving” represented in a production rules formalism. In such systems, the *Obs* are observations of learner's correct and incorrect results, the background knowledge *BK* is a representation of what the learner has supposedly understood. The induced hypotheses *Hyp* are production rules that explain the *Obs* using BK. These induced hypotheses are often used by the system to repair the learner’s misconceptions. Nevertheless, from a given set of *Obs* about the learner, the number of hypotheses that can be induced is potentially very large (Talbi & Joab, 1991). Moreover, the nature itself of the Hypotheses that satisfy the above-mentioned entailment varies from maximally specific to maximally general.

The problem of selecting appropriate hypotheses amongst all possible ones is difficult. This problem is in fact inherent to the use of induction techniques (Mitchell, 1980). The "Bounded user modeling" developed in IMPART (Elsom-Cook, 1988) is an approach that addresses one aspect of this problem. IMPART is a tutoring system for discovering the LISP language, the *Obs* are events occurring in the environment. The system builds for each event the maximally-specific conjunctive description and maximally-general conjunctive description (Hyp). These two descriptions can be regarded as lower and upper bounds on what the learner could infer from this example. However, selecting the appropriate inductive Hypotheses and narrowing the distance between the upper and lower bond is also a difficult problem. IMPART uses domain-specific heuristics to restrain such search space but does not provide general-purpose solution. In Machine Learning any basis for choosing one hypothesis over another, other than strict consistency with the observations, is more generally refereed to as *learning bias* (Mitchell, 1982). We shall also refer to the learning bias as *Bias* in the rest of the chapter.

Machine Learning to simulate human learning

Without a strong learning bias, inductive learning processes may yield to a very large number of hypothesis (Mitchell, 1980). This might explain why various learners may infer different procedures from the same provided examples of knowledge (VanLehn, 1991). CASCADE (VanLehn, 1993) exploits the fact that human learning does use induction and therefore that induction techniques may be use to simulate human learning processes. CASCADE's main principle is to use learning techniques to "detect gaps in its knowledge and fill them" . This system aims at *creating pseudo-students* " for formative evaluation during the instructional process... they simulate human learners learning from the given instruction" (VanLehn, 1991). In this approach, the *Facts* are the instruction material, the *BK* represent a type of learner's way of reasoning and the *Hyp* represent the various inferences that learners can make from the facts using the given *BK*. This approach is very useful for cognitive simulations of human learning and could provide some support to the co-learner approach (Van Lehn, Ohlson , & Nason, 1994).

Machine Learning to support collaborative guided discovery tutoring

We present in this chapter another type of MLT integration in a specific type of ITS: Guided Discovery Tutoring Environment (Elsom-Cook, 1990). We propose to use induction techniques to make the system learn concepts on the same *Facts* that are presented to the learner using as background knowledge *BK* a representation of the learner's prior knowledge. In this context, *Facts* represent some examples or observations of a given concept to be learned, and *BK*, the learner's prior knowledge, is provided by the Student Model. The set of *Hyp* produced by the system is then used to build collaborative interactions with the learner. In Guided Discovery Tutoring Environments, the learner has a greater control on the interactions with the system than in "classical" ITS. In fact, the Student Model, mostly used by ITS for choosing the nature of the teaching style, becomes in GDTE less important w.r.t. its psychological validity than w.r.t. its ability to yield plausible interactions.

In MEMOCAR, an ITS for Chinese characters by non native learner, we use induction in a context where *Facts* are a given set of Chinese characters, *BK* represents the learner knowledge about the characters description and *Hyp* are possible relations between characters that can be learned from the facts.

Although appealing in theory, there are many questions brought up with such use of inductive techniques: How such a system does *select* amongst all the possible inferences ? How to represent learner's prior knowledge so as to infer *psychologically credible* material ? How the system does *build* interactions using inferred materials ? Does the performance of today's induction systems support such interactions ? These questions are difficult ones. In this chapter, we propose some answers to these questions.

The approach we support is somewhat related to the model proposed by Gilmore and Self regarding the role of Machine Learning in ITS (Gilmore & Self, 1988) and the one presented by Elsom-Cook regarding the variability in the tutoring style (Elsom-Cook, 1988). We propose, for the sake of clarity, to classify the different approaches of induction techniques integrated within ITS in four different classes based on the type of interaction they are used for, each one corresponding to a given teaching style (See Table 1 and Table 2). The two first ones are somewhat related to maintain a Student Model whereas the last ones are related to the building of cooperative interactions. With regards to tutoring styles, these approaches range from the most constrained to the least constrained interactions: “exercises and tests”, “guided familiarization”, “guided discovery” and “unconstrained discovery”.

Table 1
Four interaction types and associated induction components

Integration	Interaction type	Comp.	Component Nature
Maintaining a Student Model	Exercises & Tests	*Obs*	**Learner's Results**
		BK	Knowledge on the Domain
		Bias	Selected by the system
		Hyp	**Misconceptions**
	Guided Familiarization	*Obs*	**Learner's exploration**
		BK	Knowledge about learner
		Bias	Selected by the system
		Hyp	**Knowledge acquired by the learner**
Building Cooperative Interactions	Guided Discovery	*Obs*	**Examples of concepts**
		BK	Learner’s prior knowledge
		Bias	Selected by the system
		Hyp	**Concepts' descriptions**
	Unconstrained Discovery	*Obs*	**Examples of concepts**
		BK	Selected by the learner
		Bias	Selected by the learner
		Hyp	**Concepts' descriptions**

Table 2

Four interaction types and associated example of systems

Interaction type	Systems Exemplifying such type of integration
Exercises & Tests	PIXIE (Sleeman, 1983)
Guided Familiarization	IMPART (Elsom-Cook, 1988)
Guided Discovery	MEMOCAR (guided mode)
Unconstrained Discovery	MEMOCAR (discovery mode)

In the following section we will emphasize the third and fourth types of interaction, guided and unconstrained discovery, by presenting MEMOCAR.

AN ITS FOR CHINESE CHARACTERS

Chinese Characters Features

Chinese is totally different from other languages as it is a non-alphabetic and non-phonetic language. A Chinese sentence is a set of characters. A character can be a word by itself or a component of word. A Chinese character is primarily defined by its written form (Bellassen, 1989). However, to memorize a Chinese character, three main type of knowledge, apparently unrelated, have to be memorized: its meanings, its pronunciations and how it is written. For example, the Chinese character indexed 4662 in Chinese dictionaries means *feelings* or *situation* (depending on the context), is pronounced qi2ng (this notation, called Pinyin, is one of the three main representation of Characters pronunciation) and written «È (See Figure 1).

We shall call a Chinese Character Feature (**CCF**), any type of information or *feature* that has to do with Chinese characters. Any CCF has, within any given character, one or more associated values (e.g. the CCF *Meaning* has two values in the character «È: feelings and situations). The CCF may naturally be structured along the more-general-than dimension (See Figure 3). *Meaning, Pronunciation* and *Written Form* are three CCF that may be termed as the three main CCF inasmuch as they represent the most important features of a Chinese character (see Figure 1).

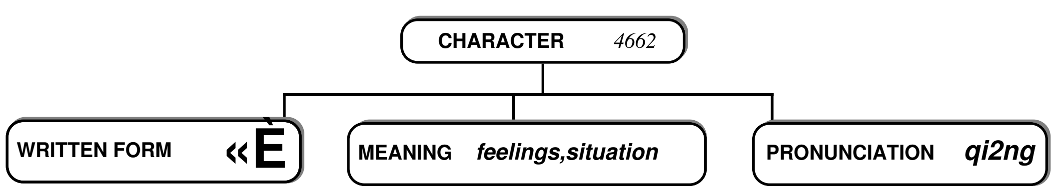

Figure 1: The three main Chinese Character Features.

Considering that the identification of about 3000 characters is a pre-requisite to understand Chinese newspapers, it clearly appears that learning Chinese characters would require an extraordinary amount of memory if characters were to be remembered as an arbitrary association of an image, a sound and an idea. In fact, the task of memorization is greatly facilitated by being familiar with other CCF besides these three main CCF (Figure 1). Let us introduce for example the CCF called *Element*. This feature is less general than the *Written Form* one. It contains the list of specific sub-components of the character that are called *Element*.

For example, the character «È can be described as containing the two *Elements* ,‡ and «‡ which are respectively identified as “radical 61” and “radical 174” in the Kang Xi dictionary (Zhang, 1979) which counts 214 radicals. Moreover, from the fact that the radical ,‡means *heart* and that «‡ is pronounced *qing*, inducing that ,‡ rather contributes to the global meaning while «‡ rather contributes to the pronunciation of the character may help memorizing the composed character. In each character, one element is called the *key*. In the character «È the *element* ,‡is the *key*. Key often provides a hint on the meaning of the character itself; they are mostly used to index dictionaries of characters.

The didactic of teaching Chinese as a second language has taken many years to acknowledge that learning Chinese characters themselves (as oppose to the understanding of words and sentences) was more than just rote learning (Bellassen, 1989). The new method of teaching Chinese characters introduced by Bellasen, suggests that the “puzzle” aspects of Chinese characters, briefly presented above, can only be appreciated with a good understanding of the various CCF (Bellassen & Pengpeng, 1991). Clearly inspired by the Bellassen method of teaching Chinese characters, we have made the hypothesis that *getting familiar with the values of the CCF and the CCF themselves is a key aspect to an efficient memorization of the characters* .

If numerous studies have tackled various cognitive aspects of Chinese as a first language (Liu, 1988), the work done on memory processes of Chinese as a foreign language are very scarce. A research group is currently working on the cognitive approach presented in this chapter. This group, called CDI (Chinese, Didactic and ITS), is mainly concerned with the validation and improvement of the pedagogical choices made in MEMOCAR; that is, the exploration environment and the kind of interactions developed. The motivation for building MEMOCAR was to offer a framework for validating the cognitive hypothesis mentioned above and to provide a system to be used by learners to consolidate their knowledge acquired during the first year under-graduate studies of Chinese at University Paris VII (Bellassen, 1989). Three main types of activity are identified in MEMOCAR: familiarization, collaborative discovery and exercises. The stage of familiarization is supported by exploration of hyperdata whilst collaborative discovery and exercises' diagnosis are supported by MLT. After presenting these three activities, we give a more technical description of the MLT used in MEMOCAR.

Familiarization with the CCF in MEMOCAR

This chapter being focused on MLT integration, we shall only briefly present the *familiarization* process developed in MEMOCAR which does not make use of MLT. Our approach to this activity consists in letting the learner explore the CCF hierarchy and each CCF associated values. This exploration of CCF allows the learner progressively to:

- be acquainted with CCF (see Figure 3). This means to progressively get familiarized with all the different features of a character. For example, a character's "average *Frequency* in written Chinese documents" is an interesting feature. Indeed, less frequent characters are somewhat less critical to memorize than frequent ones.

- get familiarized with CCF values (see table 3). Many CCF have values that may be represented as hierarchies. The *Initial* CCF for example corresponds to the initial of the pronunciation. The character «È whose pronunciation is written qi2ng has "q" as *Initial* and "ing" as *Final*. It is interesting for the learner to note that "q", "j", and "x" are phonetically closed. They are called palatals (See Figure 2).

- understand the CCF hierarchy (see Figure 3). A Chinese Character may be represented as a structure. The CCF hierarchy provides an insight of how the information about a character is structured.

- get familiarized with specific CCF values: characters that are homophone, homograph and synonym to a given character. Like in other languages, different Chinese characters may share a same meaning (synonyms). One particular aspect of Chinese characters is that a character may share the same pronunciation with other characters (homophones), and that a character may have various pronunciations (homographs).

Table 3

A few CCF and CCF values for the characters «È and æ∏

CCF	CCF values	CCF Values
WRITTEN FORM	«È	æ∏
CLASS	Ideo-phonogram	Ideo-phononogram
KEY	, ‡ ‡	«‡
KEY'S POSITION	West	East
TYPE	Character	Character
COMBINABILITY	High	Very low
FREQUENCY	High	Very low
INITIAL	q	j
FINAL	ing	ing
TONE	2	4
GAP		

ELEMENTS	,‡ «‡	¡¢ «‡
ELEMENTS-POSITION	,‡-W «‡-E	¡¢-W «‡-E
MEANINGS	feelings, situation	peace

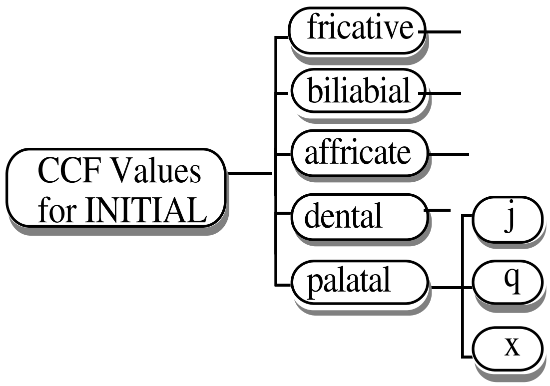

Figure 2: Part of the hierarchy representing values of the CCF *Initial*

In MEMOCAR, like in many ITS for Chinese characters (Fafiotte, 1990), characters are presented to the learner as hyperdata. However, in MEMOCAR, texts, images and sounds related to characters are organized around the notion of CCF. Figure 3 shows the CCF used in MEMOCAR, presented in the formalism of conceptual graphs (Sowa, 1984). On this figure, each box corresponds to a CCF; "caract" stands for "is characterized by" and described a CCF that is attached to another, "composed of" characterize a CCF that is composed of another less general CCF.

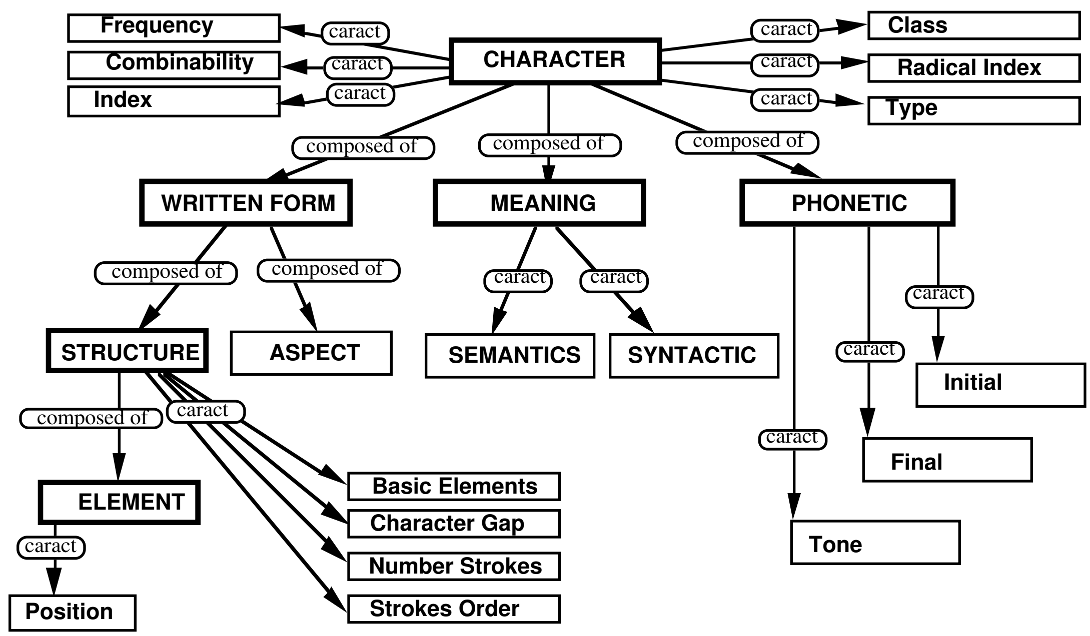

Figure 3: The Chinese Character Features' hierarchy.

What the learner knows about CCF, CCF structure and CCF values, is called the *CCF learner language* (CCFLL). This CCFLL is defined as a subset of the global CCF hierarchy and a subset of CCF values. During the CCF familiarization stage, characters are only presented to the learner using the CCFLL, which is initially composed of the three main CCF (Figure 1). We let the learner decide by himself during the interaction when he desires to discover a new CCF (which means to add a new CCF to his language) or when he desires to discover more about the values associated to a given CCF (which means to add a new CCF value for a known CCF to his language). Thanks to the CCFLL the learner may *compare* characters. This activity is supported by the second type of interaction described hereafter and proposed by MEMOCAR: discovering similarities between characters.

Guided discovery of similarities between characters in MEMOCAR

As Pinker says about language acquisition in general “...Children clearly must notice similarities holding across many pairs of items and abstract such similarities out as general rules” (Pinker, 1990). The combination of familiarization and discovery of similarities between characters has four main didactic interests:

- to improve characters' acquisition: making the learner search for similarities and dissimilarities between characters (eventually build categories of characters by maximizing within category resemblance and minimizing between-category similarity (Ahn & Medin, 1992)) .

- to suggest to the learner an analytical approach to acquire new characters constructively. That is, by analyzing a given character using the CCF he is familiar with, and comparing it with other characters.

- to make the learner aware of potential confusion between characters that have CCF with similar values (for example between characters that have a similar *written form* ≈£ ƒÎ ∏… ŒÁ (Wu, 1991)).

- to give to the learner a certain autonomy when encountering unknown characters by suggesting a probabilistic approach for guessing CCF values. The learner may indeed use the results of his experiments in detecting similarities for guessing CCF values of unknown or forgotten characters (for example when finding a character containing the radical «‡, guessing that its pronunciation ends with "ing" (see Figure 7)).

The process of discovering similarities between characters is seen in MEMOCAR as a *collaboration* between the learner and the System. Part of the learner's motivation is to "discover" interesting correlation between characters (i.e. mnemonics) using CCF so as to ease character memorization. The system's goal is to teach the learner strategies to find adequate descriptions of characters to discover useful correlation. Two modes are available to the learner for discovering similarities: a *guided mode* and a *discovery mode*. The learner's role in the collaboration is first to choose a pair of CCF in a pre-defined set (in guided mode) or to imagine (in discovery mode) an ordered pair of CCF whose correlation might be of interest. This ordered pair of CCF is called hereafter the *skeleton*. The skeleton is a *Bias* that represents the specification of the CCF to be mapped (see Table 4).

The second step is to define the subset of characters to be analyzed. To do so, the learner either chooses a pre-defined set of characters (in guided mode) or selects a subset of characters (in discovery mode) to analyze. The role of the system is to detect similarities between characters within the selected set that have the skeleton structure. However, the system must only use the CCFLL. Here follows a scenario of an interaction in discovering similarities in MEMOCAR:

- The learner chooses a pair of CCF amongst a set proposed by the system (for example *S1*: *written form ⇒ pronunciation*) or chooses any two CCF amongst the ones he has already acquired (see Table 3 below). The first CCF is called condition-CCF and the second conclusion-CCF. A skeleton expresses that the similarities to be looked for shall be of the form: <u>"If condition-CCF=?x then conclusion-CCF=?y"</u>. In fact, acceptable conditions will use any conjunction of CCF known by the learner that are at a lower level in the CCF hierarchy than condition-CCF and conclusion-CCF (for example if the condition-CCF *written form* is selected, the CCF *Elements* and *Gap* are also acceptable as condition-CCF if already acquired by the learner (see Figure 3)).

Table 4

Eight pre-defined Skeleton for Similarity Detection

CCF-CONDITION↓	CCF-CONCLUSION→	PRONUNCIATION	WRITTENFORM	CLASS	CHARACTER
PRONUNCIATION			S12	S11	S9
WRITTENFORM		S1		S10	S8
CLASS		S2	S3		S7
CHARACTER		S4	S5	S6	

- The learner then chooses a subset of characters amongst a selection proposed, or selects characters with specific CCF values. In the exploration corpus used, called CD980 (see last section), a selection "CCF Element=«‡" produces the following subset of eleven characters: «È«Â«Î«‡«Á«Ëæ´æ≤æ∏≤¬µÂ with the respective pronunciations qing, qing, qing, qing, qing, qing, jing, jing, jing, cai, dian.

- The system then performs the detection of similarities between characters using the chosen skeleton. One class of results is concerned with logical results that are always true similarities observed in the subset studied (for example “For characters in which there is the Radical «‡ and Radical's position is east then Final =ing and *Initial* =palatal“. Another class is concerned with statistical results (for example: “For characters in which there is the Element «‡*,* in 80% case *Final* =ing and *Initial* =palatal “ (see Figure 4)

- The resulting concepts learned by the system are then used in different ways. In *guiding mode*, the results are used to build a Socratic-type dialogue like about conjectures on the characters (Stevens & Collins, 1977). The dialogue allows the system to guide the learner to consider the CCF that are best fitted to describe a given group of character. The ultimate goal of the system is to teach the learner strategies to analyze characters. On the other hand, in *discovery mode*, the resulting concepts are directly submitted to the learner; the learner tries out his strategies for finding the most pertinent conjectures.

Exercises, Diagnosis and Student Model in MEMOCAR

There is an abundant literature on possible exercises to test Chinese characters memorization (Zheng, 1990): character reconstruction, modified characters, character identification in a group of characters, etc. However, we have chosen to concentrate our effort on extensive diagnosis of the supported exercises. The six following types of tests are mainly used:

- given a subset of characters’ meaning, the learner has to provide for each character its pronunciation (and vice versa),
- given a subset of characters’ written form, the learner has to provide for each character its meaning (and vice versa),
- given a subset of characters’ pronunciation, the learner has to provide for each character its written form (and vice versa).

In existing ITS for Chinese Characters, like EMICW (Castaing, 1992), the system dictates a character and requests the learner to draw the character using basic strokes. The explanations regarding dictation errors are mainly based on graphical resemblance with other characters. By contrast, MEMOCAR does not make preconceptions regarding the reasons for learner errors (e.g. like focusing on graphical resemblance), but rather attempts to induce them. MEMOCAR's diagnosis consists in extracting patterns of errors within a set of errors. Let us consider for example the first test where characters' meaning is provided by the system and their written form requested. For example (see Table 5 hereafter), the meaning of the character C1=¬Ì (horse) is given to the learner and the learner provides as an answer the written form of a character C'1.

Table 5
Three dictated characters and erroneous answers

CHARACTERS		CCF	MEANING	PRONUNCIATION	TONE	WRITTEN FORM
DICTATED CHARACTER	C1	...	**horse**	ma	3	¬Ì
ANSWERED CHARACTER	C'1	...	hemp	ma	2	È
DICTATED CHARACTER	C2	...	**ear**	er	3	∂·
ANSWERED CHARACTER	C'2	...	blood	er	4	∂-
DICTATED CHARACTER	C3	...	**study**	xue	2	ß
ANSWERED CHARACTER	C'3	...	snow	xue	3	©

After a full exercise consisting in the dictation of several characters, the system will generate a list of examples of errors (see Table 6), corresponding to a distance between the erroneous character and the correct character. These examples are described with attributes that correspond to a distance between two values of a given CCF. *Dmeaning* for example is a Boolean that equals 0 if both CCF values of C1 and C'1 are identical, and equals 1 else.

Table 6
Examples of distances between erroneous and correct characters

EXAMPLES	DCCF	DMEANING	DPRONUNCIATION	DTONE	DWRITTEN
E1=DISTANCE(C'1,C1)	...	1	0	-1	1
E2=DISTANCE(C'2,C2)	...	1	0	+1	1
E3=DISTANCE(C'3,C3)	...	1	0	-1	1

From these examples, the system extracts *similarities between errors* in the form of If-Then rules such as: If (*DMeaning=1* and *DWriten =1)* Then (*DPronunciation=0* and *DTone* $\leq$*1).* This rule stands for "In all dictated characters if there is an error on the written form, the pronunciation of the given character is only different by its tone". Such rule is very informative. It gives an example of a learner that produces wrong characters whose pronunciation are only slightly different in tone from the dictated character. Such rules are used to update the misconception's part of the Student Model.

The Student Model of MEMOCAR is composed of two different components. The first component of the Student Model in MEMOCAR represents the belief on the learner knowledge of each characters with respect to their three main CCF. In the context of MEMOCAR, the pedagogical objective is to acquire 420 characters. Thus, for each learner and for each of the 420 characters, pronunciation grade, semantics grade and written form grade are stored. This matrix is used by the system to select the characters to be used for further tests. This part of the model is rather primitive, it may be viewed as the learner's profile.

The second component of the Student Model is more qualitative and describes the learner's knowledge using CCF. It includes the CCFLL and misconceptions related to the different tests that the learner performs. These misconceptions are represented as production rules learned by the system as described above.

Selecting an Appropriate Concept Learning Technique

We describe now more technically the MLT integrated to MEMOCAR. To support the automatic detection by the system of similarities between characters and between learner errors, we had to design a tool which could take as inputs a subset of Chinese characters (Obs), a subset of CCF Values (BK), a skeleton of a relation to be extracted (Bias) and give as outputs logical and approximate rules expressing relationships between CCF (Hyp). Moreover, in order to be used in a real interaction, such tool had to offer good performances (in terms of response time, typically a few seconds). We have not used classical decision trees, although extremely efficient, because they do not allow to express adequately the learning bias (what has been described above as the skeleton). The system we have adapted to our needs is CHARADE (Ganascia, 1987). It is a symbolic induction system that extracts logical and approximate rules expressing relationships between attributes from a set of examples described using a set of attributes (in our case Chinese Characters or dictation errors).

CHARADE is based on the use of two distributive lattices, one for the learning set, i.e., the set of examples, and one for the description space. The properties of the lattices are used in the learning process to facilitate the detection of similarities and to limit the search cost. Concretely, a similarity corresponds to a correlation empirically observed in the learning set. If all the examples that have a conjunction of descriptors called d1 (e.g. *Element*=«‡) in their description, also have the descriptor d2 (*Final*="ing"), it is possible to induce that d1 implies d2 in the learning set. The principle of induction used in symbolic learning consists in a *generalization* of these relations to the whole description space. These relations must be detected. Considering the example presented on Table 3, when CHARADE compares the characters: æ∏ pronounced "**j**ing" and «È pronounced "**q**ing", it finds that they have in common: the CCF *Gap* (describing the kind of rift between Elements) with the value ∏ coding a left/right rift, *Elements*=«‡, *Final*="ing" and *Initial*="palatal" (palatal is a generalized value of "q" and "j" in the *Initial* hierarchy known by the learner, see Figure 4).

The algorithm to explore the search space goes from general to specific. So, the exploration goes from the bottom of the descriptors' lattice to its top respecting the partial order relation. The main difficulty consists in the large number of potential regularities that are useless. It is important to note that the CHARADE system makes use of logical constraints to restrict the logical implications it generates (Ganascia, 1987). As far as these constraints represent the equivalence between rules, their implementation permits us to suppress useless rules. This avoids redundancies and at the same time it limits the exploration of the description space.

One of the main advantages of CHARADE, as compared to other induction systems, is to have most of the *learning bias* explicit (Ganascia, 1991). CHARADE takes as inputs a language description which defines all attributes (e.g. CCF) , their type (unordered set, ordered set, hierarchies, etc.), their domain (e.g. CCF values, see Figure 4) and a set of training examples (e.g. a subset of CD420). The induction in itself takes advantage of the lattice structure of the search space to produce all the logical IF-THEN rules that can be expressed with some conjunctions of elements of the descriptions. The default option is to only produce the rules that do not have any exceptions: a rule does not need to cover all the examples to be produced, *but if at least one of the examples contradicts the rule*, it will not be produced. There is also a "statistical" option which produces rules that have exceptions, but it has not been used in this study. The following figure summarizes the different knowledge sources used by the CHARADE-based tool that we have developed.

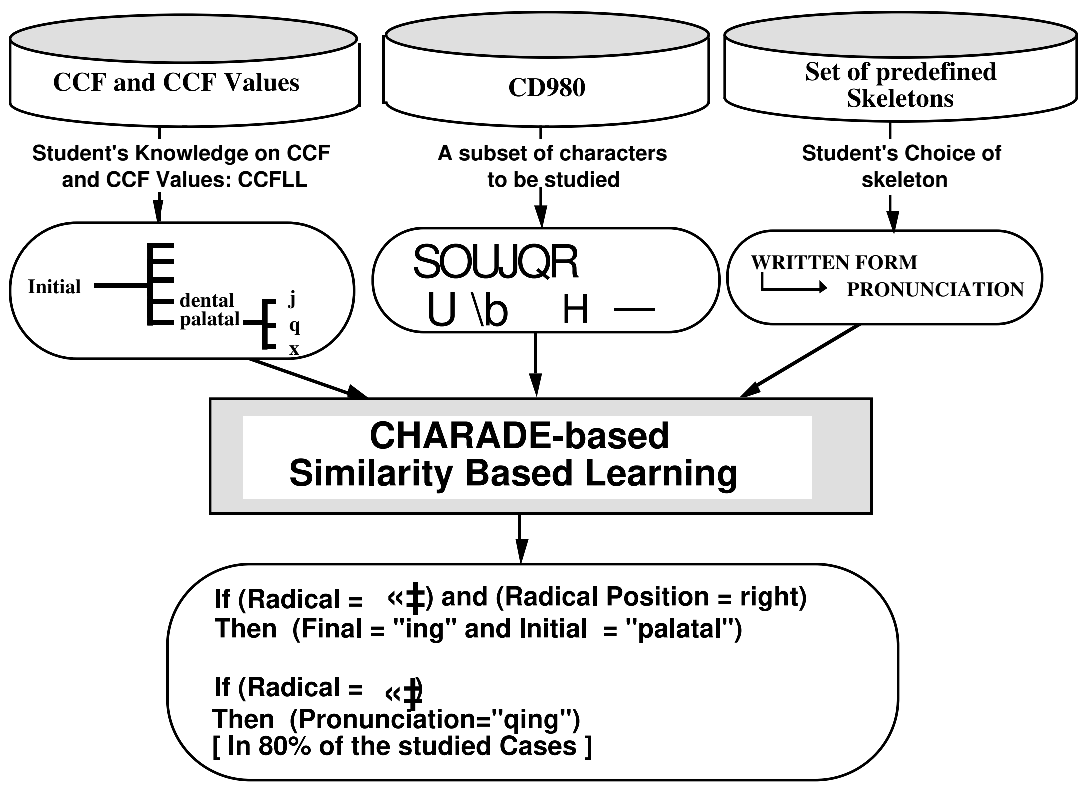

Figure 4: The different knowledge sources used by the CHARADE-based tool.

Globally, the induction process has some similarities with more classical ones like ID3 (Quinlan, 86). They can all be classified as Top-Down Induction Systems, since they search for more and more specific rules (by generating new nodes in ID3, and by exploring new layers in CHARADE). The main difference is that CHARADE does not restrict its exploration to a tree structure. Even if a training case has been successfully classified by a previous rule, it will continue searching for other rules for this example. The "classify-and-forget" aspect of the induction done by ID3 is therefore avoided, at the cost of a greater complexity for the exploration. But this makes CHARADE's induction more open, and openness is exactly the kind of feature of interest for GDTE.

MEMOCAR's implementation

In order to use CHARADE, we have represented the CCF as attributes and described the corpus CD420 and CD980 using these attributes. The first corpus, called CD420, represents the 420 characters from the Bellassen method. They are amongst the most frequently used characters and represent an average of 67% of the characters that can be seen in Chinese newspapers. CD420 represents the characters taught at school that the learner may memorize with MEMOCAR. The second corpus, called CD980, is more representative of the different CCF values that can appear within the totality of the Chinese characters. CD980 will be mainly used during the familiarization work. CD980 does nevertheless contain CD420. We have also defined hierarchies to represent structured CCF values (see Figure 2).

The general architecture of MEMOCAR is based on a Tutoring System Environment called COOPERE (Bournaud, Mathieu, & Zucker, 1993). The main components of this architecture are drafted on Figure 8 hereafter. The characters' pronunciation have been recorded by native speakers and Chinese fonts have been used to represent characters *Written Forms* and *Elements*.

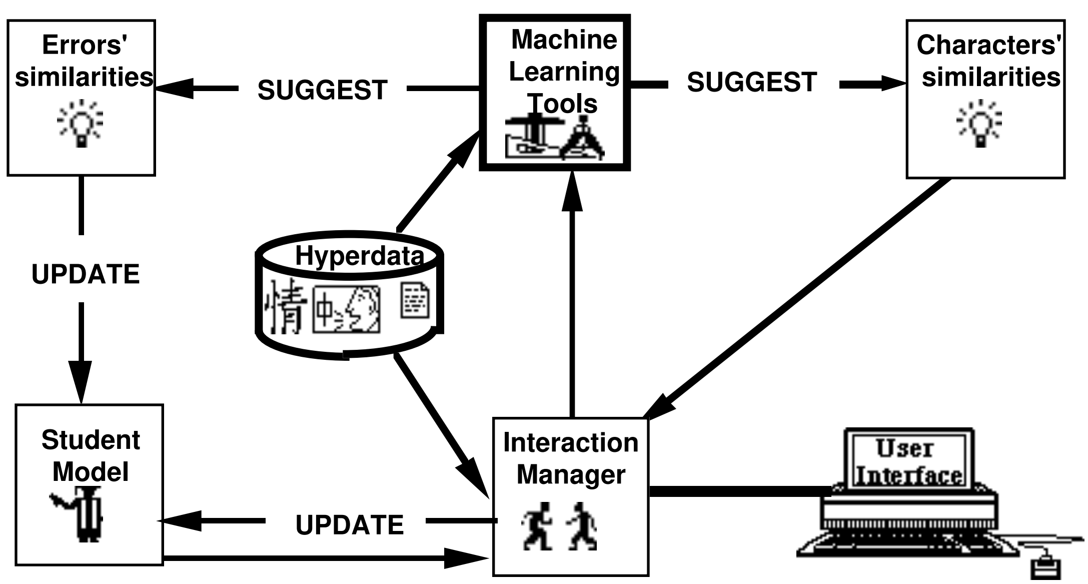

Figure 5: A view of the COOPERE-based architecture of MEMOCAR

MEMOCAR's interface is written in Hypertalk whereas the CHARADE implementation we are using, ENIGME, is developed in Symantec® C++. MEMOCAR is running on a Macintosh Quadra 950 with the Applescript® extension supporting Chinese Characters. MEMOCAR is planned to be available next year in self-service to the learners studying Chinese with the Bellassen method in the Chinese Department's library of Paris University VII.

CONCLUSION AND PERSPECTIVES

The main interest of our development is to offer the means both to *exemplify* the contributions of Machine Learning Techniques within an ITS for Chinese characters and to *analyze* the impact on memorization of our cognitive hypothesis. With MEMOCAR's availability in the Chinese department's library, we plan to assess the real benefit of our system on the learner's long term memorization of characters. The advantage of a guided discovery tutoring environment is that it allows a learner to construct new knowledge by using concepts of previously mastered knowledge. However, guided discovery environments are useful for knowledge construction by the learner if they go beyond simple exploration and include "not only browsing but also induction, deduction or problem solving tools" (Paquette, 1991). In this way, MEMOCAR integrates an interesting Machine Learning Technique that is used by the learner to discover inductively correlation between Chinese character features.

Integrating MLT to support collaborative guided discovery tutoring raises important questions. In MEMOCAR learner's prior knowledge is used to infer correlation between characters. The CCFLL that reflects the features and values that the learner is supposedly knowing allows the system to produce inferences that have a certain psychological credibility. Moreover, to *select* amongst possible inferences we have chosen to use a learning bias either selected or built by the learner. The CHARADE induction system's implementation does support such kind of integration in the sense that its response time are in seconds. The main limitation of our system is its limited use of the results provided by the MLT in guided mode. There are indeed many pedagogic uses of such material that still remain to be developed and tested. Other perspectives includes improving the quality of the inferred similarities by integrating specific concept learning techniques (Zucker & Ganascia, 1994).

REFERENCES

Ahn, A., & Medin, D. (1992). A two-stage model of category construction. *Cognitive Science*(16), 81-121.

Bellassen, J. (1989). *Méthode d'initiation à la Langue et à l'écriture Chinoises*. Paris: La Compagnie.

Bellassen, J., & Pengpeng, Z. (1991). *Perfectionnement à la langue et à l'écriture chinoises*. Paris: La Compagnie.

Bournaud, I., Mathieu, J., & Zucker, J.-D. (1993). COOPERE: Un formalisme de représentation des COnnaissances Organisées Pour l'Explication, la Résolution et l'Enseignement. In *Environnements Interactifs d'Apprentissage avec Ordinateurs*, (pp. 77-89). Cachan: Eyrolles.

Elsom-Cook, M. (1988). Guided discovery tutoring and bounded user modelling. In J. Self (Eds.), *Artificial Intelligence and Human Learning* (pp. 165-178).

Elsom-Cook, M. (1990). Introduction. In *Guided Discovery Tutoring: A Framework for ICAI Research* (pp. 4-23). London: Paul Chapman Publishing.

Ganascia, J.-G. (1987). CHARADE: A rule System Learning System. In *Proceedings of the tenth International Jointed Conference in Artificial Intelligence*, Milan.

Ganascia, J.-G. (1991). Deriving the Learning Bias from Rule Properties. *Machine Intelligence*(12).

Gilmore, D., & Self, J. (1988). The application of machine learning to intelligent tutoring systems. In J. Self (Eds.), *Artificial Intelligence and Human Learning* (pp. 178-196).

Kono, Y. (1993). *THEMIS: A nonmonotonic inductive student modeling system* (AI Technical Report No. AI-TR-93-3). Osaka University.

Langley, P., & Ohlson, S. (1984). Automated cognitive modeling. In *Proceedings of the National Conference on Artificial Intelligence*, (pp. 193-197). Austin, Texas.

Liu, I. (1988). *Cognitive Aspects of the Chinese Language*. Hong Kong: Asian Research Service.

Michalski, R. S. (1983). Learning from observation: conceptual clustering. In Y. Kodratoff & R. Michalski (Eds.), *Machine Learning: An artificial intelligence approach* San Mateo: Morgan Kaufmann.

Michalski, R. S. (1991). Toward a Unified Theory of Learning : An outline of Basic Ideas. In *Proceedings of the First World Conference on the Fundamentals of Artificial Intelligence*, (pp. 357-373).

Mitchell, T. (1982). Generalization as Search. *Artificial Intelligence Journal*, *18*, 203-226.

Mitchell, T. M. (1980). The Need for Biases in Learning Generalizations. In J. W. Shavlik & T. G. Dietterich (Eds.), *Readings in Machine Learning* Morgan Kaufmann.

Paliès, O., Caillot, M., Cauzimille-Marmèche, E., Laurière, J.-L., & Mathieu, J. (1986). Student Modeling by a Knowledge-based system. *Computational Intelligence*(2), 99-107.

Papert, S. (1991). Situating Constructionism. In *Constructionism* (pp. 1-12). Norwood: Arlex Publishing Corporation.

Paquette, G. (1991). Discovery Tools for Rule-Based Knowledge Learning. In Lewis & Otsuki (Eds.), *Advanced Research on Computers in Education* (pp. 145-150). North-Holland: Elsevier Science Publishers.

Pinker, S. (1990). Language Acquisition. In O. Sherson (Eds.), *Language* (pp. 198-210).

Quinlan, J.-R. (1986). Induction of Decision Trees. *Machine Learning(*1), 81-106.

Sleeman, D. (1983). Infering (mal) rules from pupil's protocols. In *Proceeding of International Workshop on Machine Learning*, (pp. 221-227). Illinois: Morgan Kaufmann Publishers.

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addisson-Wesley Publishing Company.

Stevens, A., & Collins, A. (1977). The goal structure of a Socratic tutor. In *Proceedings of the national ACM conference*, Seattle.

Talbi, M., & Joab, M. (1991). *Diagnostique cognitif de l'apprenant par apprentissage symbolique* No. Rapport Interne du LIF N°91-5). University Paris VI.

VanLehn, K. (1991). Two Pseudo-students: Applications of machine learning to formative evaluation. In Lewis & Otsuki (Eds.), *Advanced Research on Computers in Education* (pp. 17-26). North-Holland: Elsevier Science Publishers.

VanLehn, K. (1993). Cascade: A simulation of human learning and its application. In P. Brna, S. Ohlson, & H. Pain (Ed.), *World Conference on Artificial Intelligence in Education*, (pp. 1-3). Edinburgh, Scotland: AACE.

VanLehn, K., Ohlson, S. & Nason, R. (1994). Applications of simulated students: An exploration. to appear in *Journal of AI and Education*.

Wu, J. (1991). *Dictionary of easily confused Chinese character*. Taipei: Pioneer Language Institute.

Zhang, Y. (1979). *Kang Xi Dictionary*. Beijing: The Commercial Press.

Zheng, Y. (1990). Hanyu Jisuanji fuzhu jiaoxue xitong ke shixian tixing de fenlei yu sheji. In *Di san jie guoji hanyu jiaoxue taolunhui (Third International Symposium on the Teaching of Chinese)*, (pp. 373-375). Beijing.

Zucker, J.-D., & Ganascia, J.-G. (1994). Selective Reformulation of Examples in Concept Learning. In W. Cohen (Ed.), *International Conference on Machine Learning,* (pp. 352-360) New-Brunswick: Morgan Kauffman Publishers.

ACKNOWLEDGMENTS

The authors thank the following colleagues for their help in the MEMOCAR's design and development: Joël Bellassen, Isabelle Bournaud, Evelyne Cauzinille-Marmèche, Vincent Corruble, Jean-Gabriel Ganascia, Lucien Marnef, Jacques Mathieu, Geber Ramalho and Jérôme Thomas.